

## CLAIMS

### *We claim*

1. A computer-implemented method for controlling networking transactions among a plurality of users populating a networking database, said networking database populated with at least user identity information and relational information between said plurality of users, said method comprising:

receiving a request from a first user to perform a transaction involving at least a second user of said networking database, said request including a parameter related to said second user;

determining a feasibility of said transaction based on at least a characteristic of said requested transaction;

when said transaction is feasible, monetizing said transaction including monetization possibilities such as variable pricing for said transaction and not charging for said transaction; and

after successful monetization of said transaction, initiating execution of said transaction.

2. A computer-implemented method as recited in claim 1, wherein said parameter includes a specific relational connection type corresponding in some way to said second user.

3. A computer-implemented method as recited in claim 2, wherein said requested transaction is a search request to find all users of said networking database which have a relational connection of said specific relational connection type with said second user.

4. A computer-implemented method as recited in claim 1, wherein said second party is a second user present in said networking database, and said first and second users are members of a first virtual network defined by having a relational connection of a specific relational connection type.

5. A computer-implemented method as recited in claim 4, wherein monetizing said transaction includes first determining a cost of executing transaction based on one or more factors including at least said specific relational connection type.

6. A computer-implemented method as recited in claim 2, wherein said feasibility is a function of at least said specific relational connection type corresponding in some way to said second user.

7. A computer-implemented method for controlling and monetizing transactions between users of a computer network, the method comprising:

receiving a request from a first user to initiate a transaction that is dependent upon a second user;

determining a type of a relational connection between said first user and said second user; and

determining a cost of executing said requested transaction, said execution cost being a function of said type of said relational connection.

8. A computer-implemented method as recited in claim 7, wherein:

the requested transaction is transmission of an email message from said first user to said second user;

the cost of transmitting said email message from said first user to said second user is free when said specific type of relational connection is a first type; and

the cost of transmitting said email message from said first user to said second user is nonzero when said specific type of relational connection is a second type.

9. A computer-implemented method for controlling transactions among a plurality of parties, said computer-implemented method comprising:

populating a database with a plurality of users, said database storing at least user identity information, relational information between said plurality of users including connection types between said plurality of users, and profile data for said plurality of users:

organizing one or more virtual networks from said plurality of users;

receiving a request from a first party to search said database based upon said connection types between said plurality of users; and

performing said search when said first party is a member of said database .

10. A computer implemented method as recited in claim 9, further comprising:  
performing said search only on data made public.

11. A computer-implemented method for controlling email transactions among a plurality of users, said computer implemented method comprising:

forming at least one virtual network from a plurality of users, the at least one virtual network being formed of member users, said member users consisting of all of said plurality of users that have a relational connection of a predefined type;

receiving from a specific party an email communication intended for delivery to a specific member user;

determining whether said specific party is a member of said virtual network; and

prohibiting delivery of said email communication when said specific party is not a member of said virtual network.

12. A computer database suitable for enabling transactions between a plurality of parties including a plurality of users, said computer database comprising:

unique identity information for each of said plurality of users;

relational information associated with each of said plurality of users, said relational information useful for determining whether any two users have a connection enabling at least one type of transaction between said two users, said relational information useful for determining a degree of separation between said two users when said two users have said

connection, and said relational information defining a type for each relational connection between two users; and

virtual network information associated with each of said plurality of users, virtual network information including an identity of a virtual network of which each specific user is a member, members of each virtual network consisting of all users having a connection enabling a transaction.

12. A connections data structure for use in managing transactions among a plurality of users populating a networking database, said connection data structure comprising:

a user id 1 element arranged to store a unique identifier for a first user found in said networking database;

a user id 2 element arranged to store a unique identifier for a second user found in said networking database;

a connection status element; and

a connection type element.

13. A connections data structure as found in claim 12, wherein a status stored in said connection status element may be one of active, pending, or closed.

14. A connections data structure as found in claim 13, wherein a connection type stored in said connection type element may be one of friend type, business type, romantic type, and family type.

15. A connections data structure as found in claim 14, where said connection type stored in said connection type element may be an arbitrary type defined by said first user.

16. A connections data structure as found in claim 12, wherein a connection type stored in said connection type element may be one of friend type, business type, romantic type, and family type.

17. A connections data structure as found in claim 15, where said connection type stored in said connection type element may be an arbitrary type defined by said first user.